



Problema 1

Temporizador

Universidade Estadual de Feira de Santana
Curso: Engenharia de Computação
TEC 499 - MI - Sistemas Digitais
Professor: Thiago Cerqueira de Jesus

Equipe

- Daniel Lucas Alves Ferreira de Jesus
- Ian Zaque Pereira de Jesus dos Santos

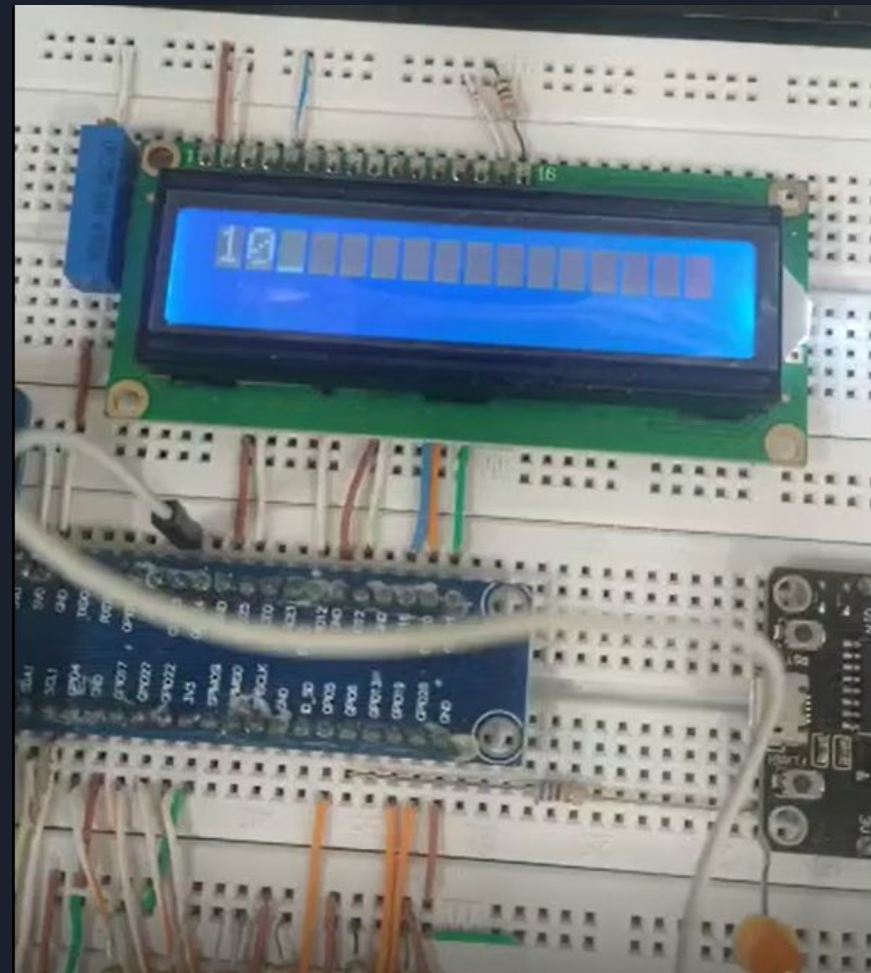


Imagem 1. Display LCD



Hardware

- Raspberry Pi Zero - Arquitetura ARMv6
- Processador Broadcom BCM 2385 1GHz, CPU single-core
- 512MB RAM
- Porta Mini-HDMI
- Porta Micro-USB
- Alimentação via Micro-USB
- Header de 40 pinos HAT-compatible
- Conector de câmera CSI
- 802.11n wireless LAN
- Bluetooth 4.0

Hardware

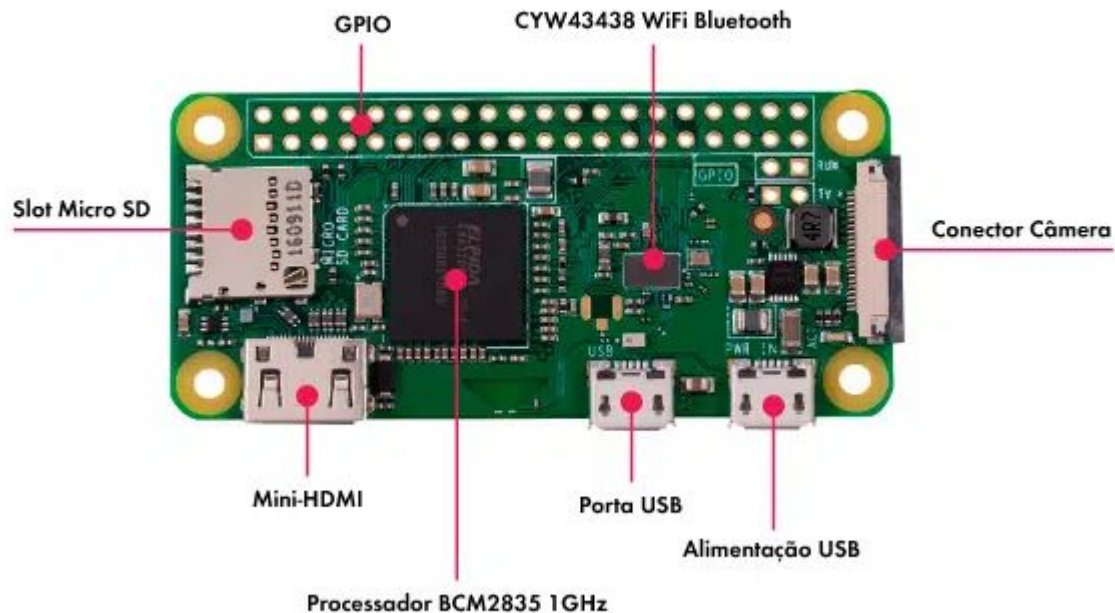


Imagem 2. Raspberry Pi Zero W.



Etapas do algoritmo

1. Mapeamento de pinos GPIO da Raspberry.
2. Definição de saída de dados.
3. Verificação de botão inicializador.
4. Contagem do temporizador.
5. Finalização de contagem.
6. Reinício do temporizador.

Macros - Funções de tempo

```
.macro nanoSleep time
    LDR R0,=\time
    LDR R1,=\time
    MOV R7, #nano_sleep
    SVC 0
.endm

.macro sleep
    LDR R0,=timespecsec
    LDR R1,=timespecsec
    MOV R7, #nano_sleep
    SVC 0
.endm
```

Imagem 3. Macros de atraso de execução.

Macros - Mapeamento GPIO

```
.macro GPIODirectionOut pin
    LDR R2, =\pin                @ address of pin table
    LDR R2, [R2]
    LDR R1, [R8, R2]
    LDR R3, =\pin                @ address of pin table
    ADD R3, #4                   @ load amount to shift from table
    LDR R3, [R3]                 @ load value of shift amt
    MOV R0, #0b111              @ mask to clear 3 bits
    LSL R0, R3                   @ shift into position
    BIC R1, R0                   @ clear the three bits
    MOV R0, #1                  @ 1 bit to shift into pos
    LSL R0, R3                   @ shift by amount from table
    ORR R1, R0                   @ set the bit
    STR R1, [R8, R2]            @ save it to reg to do work
.endm
```

Imagem 4. Macro de saída de pino.

Macros - Mapeamento GPIO

```
.macro GPIOTurnOn pin
    MOV R2, R8
    ADD R2, #setregoffset
    MOV R0, #1
    LDR R3, =\pin
    ADD R3, #8
    LDR R3, [R3]
    LSL R0, R3
    STR R0, [R2]
.endm
```

Imagem 5. Macro de saída 1 para pino.

```
.macro GPIOTurnOff pin
    MOV R2, R8
    ADD R2, #clrregoffset
    MOV R0, #1
    LDR R3, =\pin
    ADD R3, #8
    LDR R3, [R3]
    LSL R0, R3
    STR R0, [R2]
.endm
```

Imagem 6. Macro de saída 0 para pino.

Macros - Mapeamento GPIO

```
.macro setOut
    GPIODirectionOut pinE
    GPIODirectionOut pinRS
    GPIODirectionOut pinDB7
    GPIODirectionOut pinDB6
    GPIODirectionOut pinDB5
    GPIODirectionOut pinDB4
.endm
```

Imagem 7. Definição de pinos de saída.

```
.macro enable
    GPIOTurnOff pinE
    nanoSleep time1ms
    GPIOTurnOn pinE
    nanoSleep time1ms
    GPIOTurnOff pinE
    .ltorg
.endm
```

Imagem 8. Acionamento do display LCD.

Macros - Manipulação do display

```
.macro displayOff
    GPIOTurnOff pinRS
    GPIOTurnOff pinDB7
    GPIOTurnOff pinDB6
    GPIOTurnOff pinDB5
    GPIOTurnOff pinDB4
    enable

    GPIOTurnOff pinRS
    GPIOTurnOn pinDB7
    GPIOTurnOff pinDB6
    GPIOTurnOff pinDB5
    GPIOTurnOff pinDB4
    enable
.endm
```

Imagem 9. Macro desliga display LCD.

```
.macro displayClear
    GPIOTurnOff pinRS
    GPIOTurnOff pinDB7
    GPIOTurnOff pinDB6
    GPIOTurnOff pinDB5
    GPIOTurnOff pinDB4
    enable

    GPIOTurnOff pinRS
    GPIOTurnOff pinDB7
    GPIOTurnOff pinDB6
    GPIOTurnOff pinDB5
    GPIOTurnOn pinDB4
    enable
.endm
```

Imagem 10. Macro reseta display LCD.

Macros - Manipulação do display

```
.macro entrySetMode
    GPIOTurnOfff pinRS
    GPIOTurnOfff pinDB7
    GPIOTurnOfff pinDB6
    GPIOTurnOfff pinDB5
    GPIOTurnOfff pinDB4
    enable

    GPIOTurnOfff pinRS
    GPIOTurnOfff pinDB7
    GPIOTurnOn pinDB6
    GPIOTurnOn pinDB5
    GPIOTurnOfff pinDB4
    enable
.endm
```

Imagem 11. Macro ativa modo de escrita.

```
.macro write_0
    GPIOTurnOn pinRS
    GPIOTurnOfff pinDB7
    GPIOTurnOfff pinDB6
    GPIOTurnOn pinDB5
    GPIOTurnOn pinDB4
    enable

    GPIOTurnOn pinRS
    GPIOTurnOfff pinDB7
    GPIOTurnOfff pinDB6
    GPIOTurnOfff pinDB5
    GPIOTurnOfff pinDB4
    enable
.endm
```

Imagem 12. Macro escreve dígito (de 0 a 9).

Label Start

```
_start:
    @ opening the file
    LDR R0, = fileName
    MOV R1, #0x1b0
    ORR R1, #0x006
    MOV R2, R1
    MOV R7, #sys_open
    SVC 0
    MOVS R4, R0

    @ preparing the mapping
    LDR R5, =gpioaddr
    LDR R5, [R5]
    MOV R1, #pagelen
    MOV R2, #(prot_read + prot_write)
    MOV R3, #map_shared
    MOV R0, #0
    MOV R7, #sys_map
    SVC 0
    MOVS R8, R0

    setOut
    displayClear
    functionSet
    display
    entrySetMode
```

Imagem 13. Início do programa.

Labels Before e Timer

```
before:
    GPIOReadRegister pinB19
    CMP r0, r3
    BNE timer
    B before
```

Imagem 14. Verificação de botão iniciar.

```
timer:
    writeNull

    @initialValues
    MOV R11, #0b1001
    MOV R12, #0b1001

    loop:
```

Imagem 15. Execução do temporizador.

Label checkDezena e checkUnidade

```
loop:
  checkDezena:
    CMP R11, #0b1001
    BEQ dezena9
    CMP R11, #0b1000
    BEQ dezena8
    CMP R11, #0b0111
    BEQ dezena7
    CMP R11, #0b0110
    BEQ dezena6
    CMP R11, #0b0101
    BEQ dezena5
    CMP R11, #0b0100
    BEQ dezena4
    CMP R11, #0b0011
    BEQ dezena3
    CMP R11, #0b0010
    BEQ dezena2
    CMP R11, #0b0001
    BEQ dezena1
    CMP R11, #0b0000
    BNE subtrDezena
    BEQ dezena0
```

Imagem 16. Verificação da dezena do timer.

```
checkUnidade:
  CMP R12, #0b1001
  BEQ unidade9
  CMP R12, #0b1000
  BEQ unidade8
  CMP R12, #0b0111
  BEQ unidade7
  CMP R12, #0b0110
  BEQ unidade6
  CMP R12, #0b0101
  BEQ unidade5
  CMP R12, #0b0100
  BEQ unidade4
  CMP R12, #0b0011
  BEQ unidade3
  CMP R12, #0b0010
  BEQ unidade2
  CMP R12, #0b0001
  BEQ unidade1
  CMP R12, #0b0000
  BEQ unidade0
  BNE loop
```

Imagem 17. Verificação da unidade do timer.

Label subtrDezena, dezenaN, unidadeN

```
subtrDezena:  
    SUB R11, #0b0001  
    MOV R12, #0b1001  
    B checkUnidade
```

Imagem 18. Subtrator de dezena.

```
dezena9:  
    displayClear  
    write_9  
    B checkUnidade  
dezena8:  
    displayClear  
    write_8  
    B checkUnidade
```

Imagem 19. Escreve a dezena respectiva.

```
unidade9:  
    write_9  
    sleep  
    SUB R12, #0b0001  
    B checkDezena  
unidade8:  
    write_8  
    sleep  
    SUB R12, #0b0001  
    B checkDezena
```

Imagem 20. Escreve a unidade respectiva.

Seção data

```
.data
```

```
time1ms:
```

```
    .word 0  
    .word 070000000
```

```
timespecsec:
```

```
    .word 0  
    .word 300000000
```

```
fileName: .asciz "/dev/mem"
```

```
gpioaddr: .word 0x20200
```

Imagem 21. Valores de tempo e memória base

```
pinE: @ LCD Display E pin - GPIO1  
    .word 0 @ offset to select register  
    .word 3 @ bit offset in select register  
    .word 1 @ bit offset in set & clr register  
  
pinDB4: @ LCD Display DB4 pin - GPIO12  
    .word 4 @ offset to select register  
    .word 6 @ bit offset in select register  
    .word 12 @ bit offset in set & clr register  
  
pinDB5: @ LCD Display DB5 pin - GPIO16  
    .word 4 @ offset to select register  
    .word 18 @ bit offset in select register  
    .word 16 @ bit offset in set & clr register  
  
pinDB6: @ LCD Display DB6 pin - GPIO20  
    .word 8 @ offset to select register  
    .word 0 @ bit offset in select register  
    .word 20 @ bit offset in set & clr register  
  
pinDB7: @ LCD Display DB7 pin - GPIO21  
    .word 8 @ offset to select register  
    .word 3 @ bit offset in select register  
    .word 21 @ bit offset in set & clr register  
  
pinB19: @ LCD Display B19 button pin - GPIO19 - Start/Pause  
    .word 4  
    .word 27  
    .word 524288
```

Imagem 22. Valores para pinos do display e botões.



Referências

Imagem 1 a 22, exceto 2. Disponíveis em: https://github.com/ian-zaque/pbl_SD_1

Imagem 2. Raspberry Pi Zero W. Disponível em: www.filipeflop.com/blog/raspberry-pi-zero-w-com-wifi-e-bluetooth

Dados sobre Raspberry Pi Zero W. Disponível em: www.raspberrypi.com/products/raspberry-pi-zero-w